AFNI Jazzercise

Please read the following questions and use your AFNI know-how to answer them. Hints to answering these questions are available in the "Hints" handout. The answers to these questions can be found in the "Answers" handout.

1. The dataset **AFNI_data3/afni/`func_slim+orig`** contains 7 sub-bricks of statistical data. Use **`3dbucket`** to create a smaller version of this dataset that contains only the sub-bricks: #0, 3-6. Name this new dataset **some_stats**.
   Why: To understand the layout of AFNI datasets and sub-bricks. You'll probably be interested only in specific sub-bricks and not all the sub-bricks that include baseline fit statistics.

2. In directory **AFNI_data3/afni** you will find three anatomical datasets: **`anat1+orig`**, **`anat2+orig`**, **`anat3+orig`**. These datasets are 3 separate anatomical scans of a single subject. They have already been aligned. Average them together into a single dataset called **anat_mean+orig**. Notice that the result looks 'cleaner', since the noise has been reduced.
   Why: averaging reduces the noise.

3. Use AFNI's two skull-stripping programs, **`3dIntracranial`** and **`3dSkullStrip`**, to remove the skull from dataset **AFNI_data3/afni/`anat+orig`**. Name the output file from 3dIntracranial **anat_3dIntra+orig** and the output file from 3dSkullStrip **anat_3dSkull+orig**. Compare the two output datasets. Did one program do a better job at skull stripping or are the results similar? (Note: 3dSkullStrip may take a few minutes to run so be patient).
   Why: Removing the skull is useful for image registration and creating a brain-specific mask.

4. Creating and Playing with ROI Masks:
   a. The dataset **AFNI_data3/afni/`func_slim+orig`** has beta values and t-stats for 2 stimulus classes, <u>fpos</u> and <u>fneg</u>. Use **`3dcalc`** to create a mask called **PN_mask** that is **1** everywhere that both the fpos t-stat and the fneg t-stat values are greater than 4.2, and **0** everywhere else.
   Why: Combining the results in a single mask is useful for a simple conjunction analysis.

   b. Similar to part a, create a conjunction mask that is **1** wherever a>4.2 (from fpos t-stat sub-brick), **2** wherever b>4.2 (from fneg t-stat sub-brick), **3** wherever both are true, and **0** otherwise. Name this dataset **PN_mask_4+orig** (since it contains 4 values).
   Why: Using masks with values of powers of two are easier to look at all possible combinations in conjunction analysis.

   c. Use the afni GUI to display this mask, **`PN_mask_4+orig`**, so that each mask value gets its own color. What does each color mean?
   Why: The various combinations will each show a unique color that is easily visible and understood.

   d. Use **`3dROIstats`** to store the average time series from **`epi_r1+orig`** into the text file **PN_mean.1D**, where the mean is over the voxels in the mask (from part a), **`PN_mask+orig`**.
   Why: Mean time series curves for each ROI are useful for further analysis and display with other AFNI programs or with external applications like Excel and Matlab.

5. Fun with 1D files:
   a. Create three 1-column files with the numbers 1-10 in one column of the first file, 11-20 in the second file, and 21-30 in the third file. (note: you might use 2 different AFNI programs to create each file)

   b. Catenate these 3 files into one 3-column file. Call this 1D file **3_cols.1D.**

   c. Create a new file that contains columns 1, 2, 3, 3, 2,1, from part b (i.e., there will be a total of 6 columns in this new 1D file). Call this new 1D file **6_cols.1D**.

   d. Now take the 6 columns from question 7b and average them together to create a new file with a single column. Call that new file **ex_mean.1D**.

   Why: AFNI 1D programs can help to combine and analyze data from multiple voxels or masks.

6. Fun with the AFNI GUI:
   a. Open **AFNI_data3/afni/`anat+orig`** and in any one of the views (sagittal, axial, or coronal), change the gray-scale intensity range to be 500 minimum and 1500 maximum.

   b. Open **AFNI_data3/afni/`func_slim+orig`** and set the Full-F as the OLay and Threshold. Set the Threshold to F=8.0. Show only Positive values and set the color scale to show only 8 colors. Edit the color scale so that F-values between 14 and 28 are shown in lime green.

   c. View the above settings you created from part b in a sagittal slice. Make a jpeg file from sagittal slice #166 and name it **cool_slide**.

   d. Switch to <u>Talairach</u> view and Talairach to the **right fusiform gyrus**.

   e. Change the display to show 6 sagittal slices all at once, in a 3x2 montage.

   f. Can you find the AFNI Mission statement hidden in the AFNI GUI?

   Why: The AFNI GUI has the flexibility to look at your data, and it's fun once you learn how.

7. Doing Calculations in AFNI:
   a. Determine what type of data (short, float, etc) makes up dataset **AFNI_data3/afni/`func_slim+orig`**.

   Why: It's important to know how to find information about your dataset.

   b. Calculate 22.3 * 44.5 using the simple calculating program in AFNI.

   Why: AFNI provides calculation tools including simple ones that are useful in scripts.

8. Image Filtering:
   a. Smooth **AFNI_data3/afni/`epi_r1+orig`** with a 8mm FWHM filter. Name the output file **ex_blur8**.

   b. Enhance **AFNI_data3/afni/`anat+orig`** by emphasizing the minimum-valued voxels across +/- 3 voxels in the sagittal (z) direction. Name the output dataset **ex_minz3**.

c. Enhance dataset **ex_minz3+orig** from part b by removing the noise with program **3danisosmooth**. Name the output dataset **ex_aniso**. Use the **-viewer** option in this program to select the number of noise-removing iterations.

Why: Image filtering provides some powerful ways to enhance the data. Data can be changed in radically different ways. Use this power wisely.

9. Random Exercises with AFNI Datasets:

a. Open dataset **AFNI_data3/afni/anat+orig** dataset and find the spatial storage order (i.e., xyz-orientation). Re-orient it to LPI orientation and name the new output dataset **exLPI**.

Why: Data may be required by specific programs to match other data or to match a specific orientation.

b. Open dataset **AFNI_data3/afni/func_slim+orig** and create 2 separate datasets: one with the 3rd sub-brick only and one with the 4th sub-brick only. Call the former dataset **ex_fneg_coef** and the latter **ex_fneg_tstat**.

Why: Extracting specific data is useful for exporting to other software or for making the data fit in memory more easily if it's a large dataset.

c. Combine **ex_fneg_coef+orig** and **ex_fneg_tstat+orig** from part b into a single dataset called **ex_fneg**, having the fneg Coef for sub-brick 0, and fneg t-stat for sub-brick 1.

Why: Datasets can be manipulated to include only what you're interested in.

d. Convert dataset **AFNI_data3/afni/func_slim+orig** to Talairach coordinates with a $4mm^3$ resolution. Use dataset **anat+tlrc** in the same directory as the data parent to perform the transformation on **func_slim+orig**. Name the output file **func_slim4mm**.

Why: Talairach data will be $1mm^3$ by default. This resolution is often not necessary because it doesn't reflect the resolution of the EPI data. It makes processing slower and taxes memory too.

e. Locate the maximum "Full-F" stat voxel value in dataset **func_slim4mm+tlrc** and find the name of the Talairach atlas region that corresponds to that voxel's position.

Why: Finding maximum activation can be scripted or searched interactively. Atlas regions should be used as a guide. The AFNI GUI includes various atlases that cite the regions associated with any specific voxel.

f. Dataset **AFNI_data3/afni/anat+orig** was acquired sagittally and contains 124 slices. Create a new dataset that contains only slices 40-90 of anat+orig. Provide the new dataset with the prefix name **anat_40_90**.

Why: If memory or processing speed is a constraint, you can work with only part of the data.